

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 64-012339

(43)Date of publication of application : 17.01.1989

(51)Int.Cl.

G06F 9/44

(21)Application number : 62-166911

(71)Applicant : OKI ELECTRIC IND CO LTD

(22)Date of filing : 06.07.1987

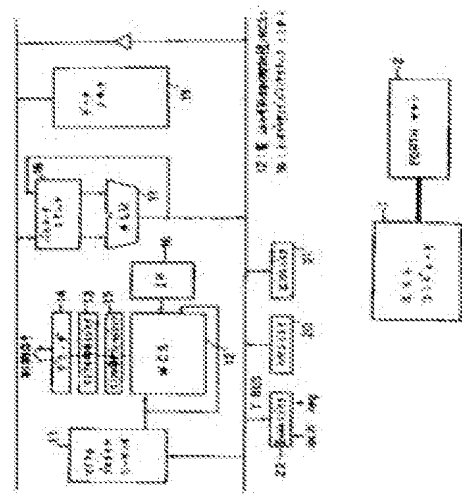
(72)Inventor : MATSUSHIRO NOBUHITO
OYA IKUO

(54) FORTH MACHINE

(57)Abstract:

PURPOSE: To realize the execution of a word and a link processing between the words and to eliminate link overhead between the words, by providing an interpretive pointer stack and a microinstruction mask.

CONSTITUTION: A FORTH machine 2 is backend machine for a host computer 1, and it is constituted so that the dictionary of a FORTH word can be formed in a WCS12 and it can be made access from the host computer 1. The address of the word to be executed is supplied from the host computer 1, and an execution result is delivered to the host computer 1. A microinstruction read out from the WCS12 to a microinstruction register 13 by a sequencer 11 supplies a control signal to every part passing a decoder 14. The interpretive pointer stack 16 is used for the designation of a word address.



⑩ 日本国特許庁(JP)

⑪ 特許出願公開

⑫ 公開特許公報(A)

昭64-12339

⑬ Int.Cl.⁴

G 06 F 9/44

識別記号

3 3 0

庁内整理番号

D-8724-5B

⑭ 公開 昭和64年(1989)1月17日

審査請求 未請求 発明の数 1 (全8頁)

⑮ 発明の名称 FORTHマシン

⑯ 特 願 昭62-166911

⑰ 出 願 昭62(1987)7月6日

⑱ 発 明 者 松 代 信 人 東京都港区虎ノ門1丁目7番12号 沖電気工業株式会社内
⑲ 発 明 者 大 宅 伊 久 雄 東京都港区虎ノ門1丁目7番12号 沖電気工業株式会社内
⑳ 出 願 人 沖電気工業株式会社 東京都港区虎ノ門1丁目7番12号
㉑ 代 理 人 弁理士 山本 恵一

明 細 書

1. 発明の名称

FORTHマシン

2. 特許請求の範囲

書き込み可能な制御記憶(WCS)を備え、ホストコンピュータによりバックアップされ、FORTH言語で記述されたプログラムを実行するFORTHマシンにおいて、

前記書き込み可能な制御記憶(WCS)内に作成されるワードアドレスを直接アクセスし、かつリターンアドレス計算を行うハードウェア構成のスタブリティブポインタスタックと、

全マイクロフィールドからワードアドレスフィールド及びフラグフィールドのみを出力するマイクロ命令マスクを設け、

ワードの実行と、ワード間のつなぎ処理を並列に実行させることを特徴とするFORTHマシン。

3. 発明の詳細な説明

(産業上の利用分野)

本発明は、FORTH言語で記述されたプログラムの高速実行を目的としたFORTHマシンに関するものである。

(従来の技術)

プログラミング言語FORTHは、図形処理、画像処理をはじめデータベースや他の言語プロセッサの開発まで広く用いられ、多くのFORTHシステムが市販されている。FORTH言語は、関数呼出しの制御に用いられるリターンスタックと、あらゆる演算のオペランドとなるパラメータスタックを持ち、実行時におけるスタックの負荷が非常に大きいという特徴がある。

以上の点に着目し、ハードウェアによるパラメータスタック、リターンスタックを設けたマイクロプログラム制御方式のFORTHマシンが開発されている。この種のFORTHマシンとしては、例えば「FORTHマシンシステムの評価」電子通信学会論文誌別刷、電子通信学会、Vol. J65-D No.3、1982年3月、p 346-353に開示されたものがある。同文献に記載のFORTHマシンは、ホ

ストコンピュータに対するバックエンドマシンであり、ホストコンピュータがシステムの起動やコンパイル、入出力処理、ファイルの管理などFORTHマシンをバックアップする形態をとっている。そしてFORTHマシンの書込み可能な制御記憶(Writable Control Storage;以下WCSという)上にFORTHディクショナリを形成し、ユーザが入力したプログラムはマイクロ命令にまでコンパイルされる。なおここでディクショナリとはワードの集合体であり、ワードは処理の単位(FORTRANのサブルーチンに相当)である。またワード(xyz)をコンパイルするとは、ワード(xyz)の入口番地を呼び先とするCALLシーケンスをディクショナリに作っていくことである。実行(ワードを実行するとは、ワード(xyz)の入口番地を呼び先としてCALLすること)をいうのは、ホストコンピュータから実行すべきワードのアドレスが与えられ、実行結果がホストコンピュータに渡されることによりなされる。パラメースタック、リターンスタック、AL

点がある。また、4マイクロサイクル以上で実行されるワードのつなぎオーバーヘッドも小さいとはいえない。

本発明は、このような従来技術の問題点を解決するためになされたものであって、FORTHコンパイラの負荷を増大させることなく、ワード間のつなぎオーバーヘッドをなくし、プログラム前編FORTHを高速に実行するFORTHマシンを提供することを目的とする。

(問題点を解決するための手段)

本発明は、書込み可能な制御記憶(WCS)を備え、ホストコンピュータによりバックアップされ、FORTH言語で記述されたプログラムを実行するFORTHマシンを対象とし、前記従来技術の問題点を解決するため、前記書込み可能な制御記憶(WCS)内に作成されるワードアドレスを直接アクセスし、かつリターンアドレス計算を行うハードウェア構成のインタプリティブポインタスタックと、全マイクロフィールドからワードアドレスフィールド及びフラグフィールドのみを

U等を並列に動作させることにより、大多数のFORTH基本ワードを1~3マイクロサイクルで実行できる。

ここで、基本ワードの実行が高速化された際問題となるのが、第10図(図中ヘッダは省略)のようなFORTH内部インタプリタによるワード間のつなぎ(ワード本体のワードアドレス(以下WAという)へジャンプし、リターンする処理)オーバーヘッドである。すなわち、基本ワードの呼出しを行うと、1マイクロサイクルの処理のためにジャンプとリターンに2マイクロ命令必要となる。

上記文献のFORTHマシンでは、1~3マイクロサイクルで完了するような基本ワードが参照された場合、コンパイル時にその処理に対応する直接マイクロコードが生成され、ワード本体に埋め込まれるようになっている。

(発明が解決しようとする問題点)

しかしながら、上記文献のFORTHマシンではワードの実行は高速になるが、コンパイラが複雑になる、コンパイルに時間がかかるという問題

出力するマイクロ命令マスクを設け、ワードの実行とワード間のつなぎ処理を並列に実行させるようにしたものである。

(作用)

本発明では、インタプリティブポインタは実行中のワード本体の最終マイクロ命令サイクルの初期に、書込み可能な制御記憶(WCS)のワードアドレスを直接アクセスしマイクロフィールドを読み出し、リターンアドレスを計算する。マイクロ命令マスクはインタプリティブポインタが読み出した全マイクロフィールドとフラグフィールドのみを出力する。そして次のマイクロサイクルでジャンプ先のマイクロ命令を実行することにより、ワードの実行とワード間のつなぎ処理がオーバーラップされ、つなぎオーバーヘッドがなくなる。従って、前記従来技術の問題点が解決される。

(実施例)

以下本発明の実施例につき詳細に説明する。

第1図は本実施例に係るFORTHマシンの内部構成を示すブロック図、第2図は本実施例の全

体構成図である。

先ず第2図により全体構成について述べると、図中1はホストコンピュータ、2はFORTHマシンである。FORTHマシン2はホストコンピュータ1に対するバックエンドマシンであり、ホストコンピュータ1がシステムの起動やコンパイル、入出力処理、ファイル管理などFORTHマシン2をバックアップする形態をとる。FORTHマシン2のWCS上にFORTHワードのディクショナリを形成し、ホストコンピュータ1からもアクセスできるようになっている。ワードのサーチはホストコンピュータ1が行う。実行はホストコンピュータ1から実行すべきワードのアドレスが与えられ、実行結果がホストコンピュータ1に渡されることによる。

次にFORTHマシン1の内部構成について説明する。第1図に示すように、FORTHマシン1はマイクロプログラムシーケンサ11、WCS12、マイクロ命令レジスタ13、デコーダ14、マイクロ命令フィールドマスク15、スタックトップにカウ

ンタが付いているインタプリティブポインタスタック16、レジスタファイル付のALU17、パラメータスタック18、データメモリ19、入力レジスタ20、出力レジスタ21及び割込みレジスタ22から構成される。

シーケンサ11によりWCS12からマイクロ命令レジスタ13に読み出されたマイクロ命令は、デコーダ14を通り各部に制御信号を与える。インタプリティブポインタスタック16は、次に実行すべきワードを指すインタプリティブポインタをハードウェアで実現するスタックであり、WCS12上に作成されるWAを格納しているメモリアドレスを指定するのに使用する。マイクロ命令フィールド f_0 が1(High)のときスタック16をポップする。 f_0 が1のときスタックトップをインクリメントする。 f_1 が1のときスタックトップがWCS12のアドレスを指定し、WCS12の内容が読み出される。 f_2 が1のときシーケンサ1のプログラムカウンタの内容をスタック16にプッシュしたのちスタックトップをインクリメントする。マイクロ

命令フィールドマスク15は、 f_1 が1のときインタプリティブポインタスタック16によりWCS12から読み出された全マイクロフィールドのうち、シーケンサ11に与えるジャンプアドレスフィールド及び f_0 フィールドのみを出力し、マイクロ命令レジスタ13に書き込むためのマスクである。なお f_0 、 f_1 、 f_2 、 f_3 については第3図により後に説明する。ALU17はFORTH各ワードの基本演算実行に用いる。パラメータスタック18は演算のオペランドとしてのスタック値を格納する。データメモリ19は、リターンスタック値、数値データの一時退避等に使用する。入力レジスタ20、出力レジスタ21、割込みレジスタ22はホストコンピュータ1とのデータのやりとりに使用する。

次に第3図によりマイクロフィールドの構成について説明する。図示の如くマイクロフィールドは、入出力関係命令、データメモリ関係命令、パラメータスタック関係命令、ALU関係命令、シーケンサ関係命令、 f_1 、 f_2 、 f_3 、 f_0 、シーケンサジャンプアドレスの各フィールドから成る。

f_1 、 f_2 は2以上の語(後述)の最後のWAとともに、 f_3 、 f_0 は原語(後述)及びDOCOL(後述)の最後のマイクロ命令とともに、コンパイル時に1(High)にセットされる。

次に第4図によりワードの構成について説明する。WCS12内に作成されるワードには大別して二つの型がある。二つの型のワードとは、

- (1) 原語(プリミティブ)
- (2) $\left\{ \begin{array}{l} 2 \text{ 次語(セカンダリ)} \\ N \text{ 次語} \end{array} \right.$

でそれぞれのWCS12内での配置、および構造を第4図に示す。図中ヘッダはワードの見出し、NEXTは原語におけるリターンを実行するもの、DOCOLは2次語の先頭にあって該DOCOLの次にあるメモリをWAとみなし、それ以降のワードを次々と実行していくように制御するもの、SEMIは2次語におけるリターンを実行するものである。NEST、SEMIの実体は f_3 、 f_2 、 f_1 により制御されるインタプリティブポインタスタック16の働きである。

原語とは、その本体がマイクロ命令で構成されるものであり、2次語とは、その本体が原語に対するアドレスポインタで構成されるものである。N次語とは、その本体が(N-1)次以下の語(含原語)に対するアドレスポインタで構成されるものである。

本実施例は、実行中のワード本体の最終マイクロ命令実行サイクルの初期に、IP16により読み出されたWAをシーケンサ11のジャンプアドレスフィールドに書き込み、IP16をインクリメント(リターンアドレスを計算)し、次のマイクロサイクルでジャンプ先のマイクロ命令を実行することにより、ワードの実行とワード間のつなぎ処理をオーバーラップさせ、つなぎオーバーヘッドをなくする。サイクルタイムはたとえば200nsecとし、WC S12は50nsecのSRAMで構成する。

次に上記実施例の動作について第5図ないし第9図を参照して説明する。

まず、2次語に対するWAのみで構成される3次語の場合の動作につき第5図ないし第7図によ

り述べる。第5図は本例の場合のワード間のつなぎを示す図(図中ヘッダは省略)、第6図はIPのマイクロステップによる変化を示す図、第7図(a)は本例の場合のマイクロステップ2, 4, 7, 12及び後述の2次語原語に対するWAが混在する3次語の場合のマイクロステップ13におけるWA、f。セットとIPインクリメントのタイミング図、第7図(b)は本例の場合のマイクロステップ11におけるWA、f。セットとIPポップのタイミング図である。

本例の場合の動作は次に示すように行われる。

(以下余白)

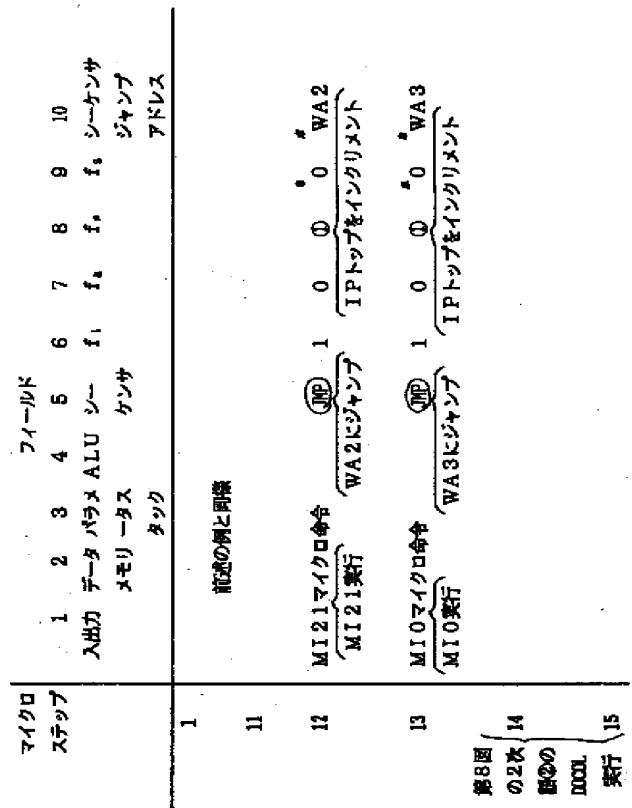
マイクロ ステップ	フィールド										ジャンプ アドレス
	1	2	3	4	5	6	7	8	9	10	
第5図の 3次語① のDCL 実行	入出力 データ メモリ	データ メモリ タック	パラメ ータス	ALU シーケ ンサ	シーケ ンサ	f。	f。	f。	f。	f。	シーケ ンサ ジャンプ アドレス
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語① のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語② のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語③ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語④ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語⑤ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語⑥ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語⑦ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語⑧ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語⑨ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語⑩ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語⑪ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語⑫ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語⑬ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語⑭ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語⑮ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語⑯ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語⑰ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語⑱ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語⑲ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語⑳ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語㉑ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語㉒ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語㉓ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語㉔ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語㉕ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語㉖ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語㉗ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語㉘ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語㉙ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語㉚ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語㉛ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語㉜ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語㉝ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語㉞ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語㉟ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語㊱ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語㊲ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語㊳ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語㊴ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語㊵ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語㊶ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語㊷ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語㊸ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語㊹ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語㊺ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語㊻ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語㊼ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語㊽ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語㊾ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	
第5図の 2次語㊿ のDCL 実行	1	2	3	4	5	6	7	8	9	10	
	1	2	3	4	5	6	7	8	9	10	

なお上記で・印はIP16によりWCS12から読み出され、マスクされてマイクロフィールドに書き込まれたWA、f₁を示す。

上記のすべてのマイクロステップで2次語、3次語のワードDOCOL、原語のマイクロ命令を実行しており、第10図に示すようなつなぎ処理オーバーヘッドが無くなる。N次語にしても同様である。

次に、2次語、原語に対するWAが混在する3次語の場合の動作につき第8図、第9図及び第7図を参照して説明する。第8図は本例の場合のワード間のつなぎを示す図(図中ヘッダは省略)、第9図はIPのマイクロステップによる変化を示す図である。

(以下余白)



上記すべてのマイクロステップで2次語、3次語のワードDOCOL、原語のマイクロ命令を実行しており、第10図に示すようなつなぎオーバーヘッドが無くなる。N次語にしても同様である。

本実施例による効果として、2次語実行における性能比較結果を次表に示す。

(以下余白)

表

A	8	1	2	3	4
4	0.41	0.56	0.65	0.70	
8	0.37	0.53	0.63	0.68	
16	0.36	0.52	0.62	0.67	
32	0.34	0.50	0.60	0.65	

A: 2次語本体中のWA数

B: 2次語から呼ばれる原語の

平均マイクロ命令数

表の数値: (つなぎオーバーヘッド無の場合の2次語実行マイクロステップ)

(つなぎオーバーヘッド有の場合の2次語実行マイクロステップ)

(発明の効果)

以上詳細に説明したように、本発明によれば、インタプリティブポインタスタック及びマイクロ命令マスクを設けたので、ワードの実行とワード間のつなぎ処理が並列に実行可能となり、FORTHコンパイルを複雑にすることなく、すべてのワード間のつなぎオーバーヘッドをなくすることができ、しかも高速にワードの実行が行えるようになる。

4. 図面の簡単な説明

第1図は本発明の実施例に係るFORTHマシンの内部構成図、第2図は本発明の実施例の全体構成図、第3図はマイクロフィールドの構成を示す図、第4図はワードの構造を示す図、第5図は2次語に対するWAのみで構成される3次語の場合のワード間のつなぎを示す図、第6図はIPのマイクロステップによる変化を示す図、第7図はWA、I、セットとIPインクリメントあるいはIPポップのタイミング図、第8図は2次語、原語に対するWAが混在する3次語の場合のワード間

のつなぎを示す図、第9図はIPのマイクロステップによる変化を示す図、第10図は従来技術によるワード間のつなぎを示す図である。

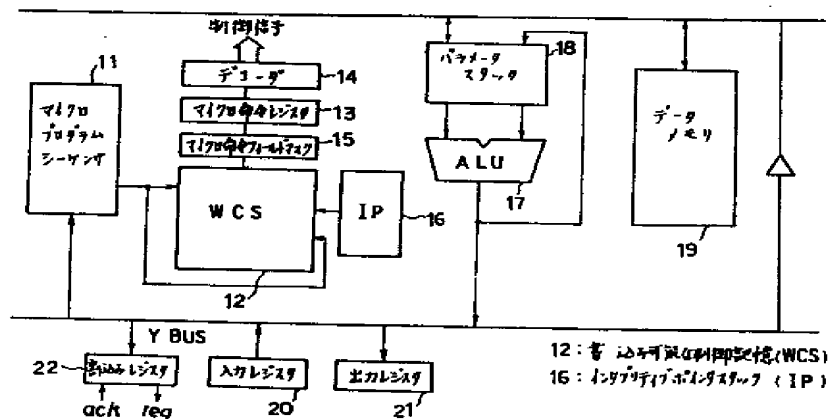
1---ホストコンピュータ、 2---FORTHマシン、 11---マイクロプログラムシーケンサ、 12---書込み可能な制御記憶(WCS)、 13---マイクロ命令レジスタ、 14---デコーダ、 15---マイクロ命令フィールドマスク、 16---インタプリティブポインタスタック(IP)、 17---ALU、 18---パラメータスタック、 19---データメモリ。

特許出願人

沖電気工業株式会社

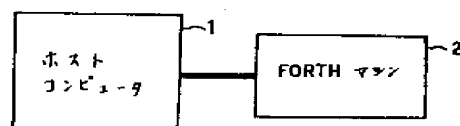
特許出願代理人

弁理士 山本恵一



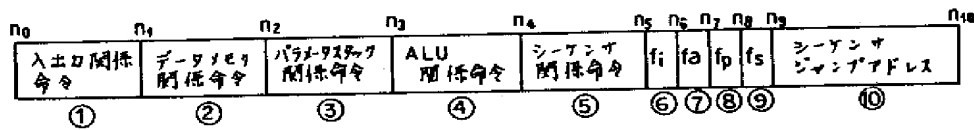
本発明の実施例に係るFORTHマシンの内部構成図

第 1 図



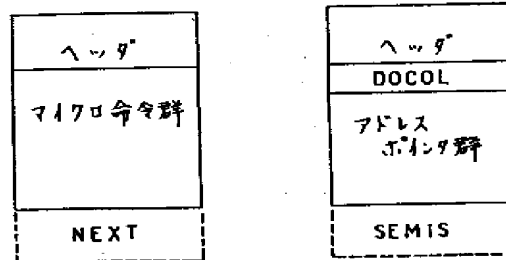
本実施例の全体構成図

第 2 図



マイクロフィールドの構成図

第 3 図

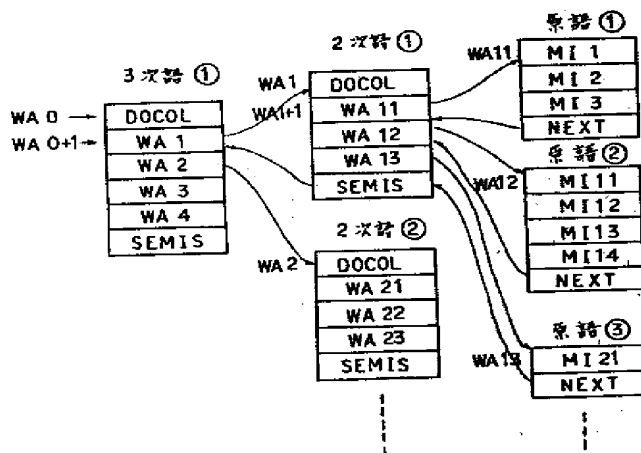


原 語

2 次 語

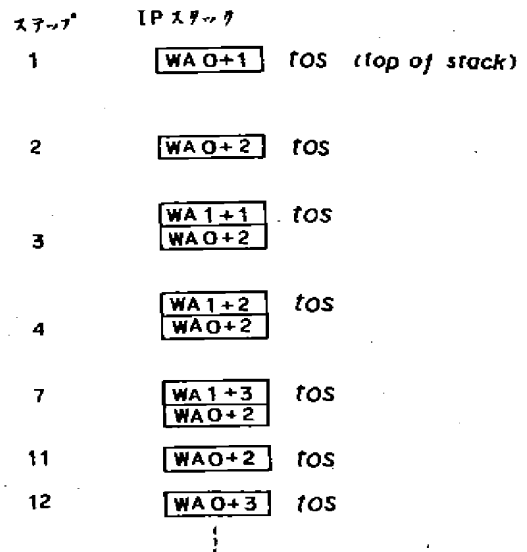
ワードの構造図

第 4 図



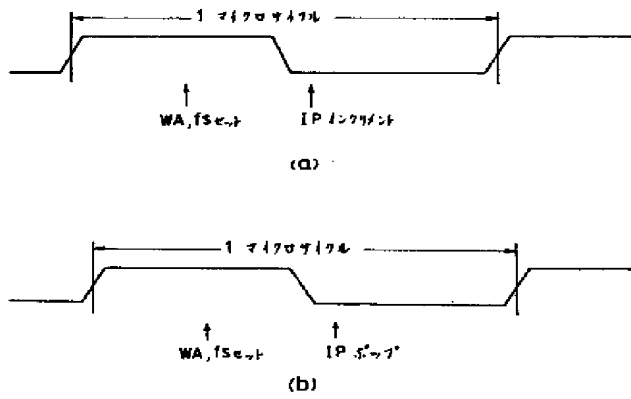
ワード間の関係を示す図

第 5 図



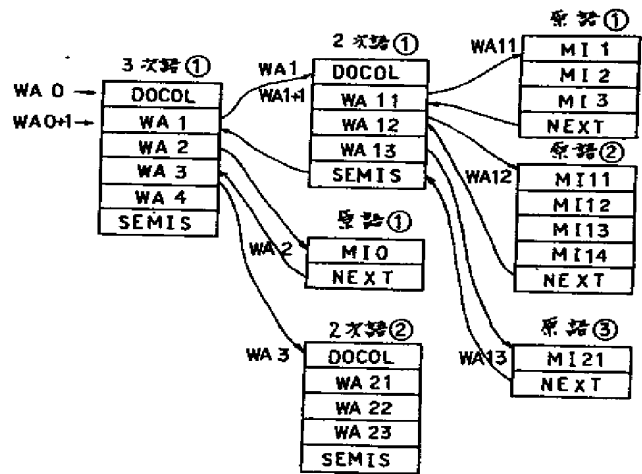
IPの变化を示す図

第 6 図



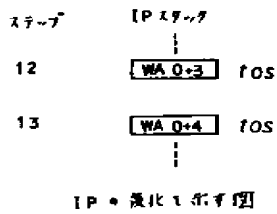
WA, fscet と IP 及び IP, fscet のタイミング図

第 7 図

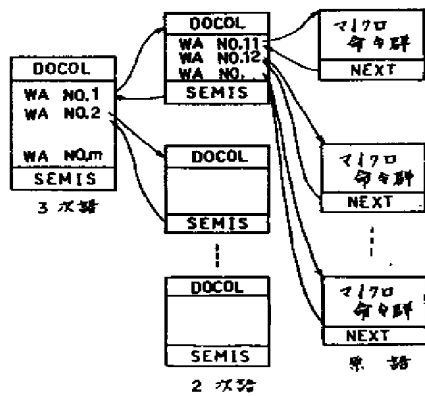


ワード間のつながり図

第 8 図



第 9 図



従来技術によるワード間のつながり図

第 10 図